ra-ma.es

DISEÑO DE PROGRAMAS - RA-MA

6-8 minutos

ÍNDICE

INTRODUCCIÓN

CAPÍTULO 1. LENGUAJE ALGORÍTMICO

- 1.1 DEFINICIÓN DE ALGORITMO
- 1.2 LA ROTACIÓN DE UNA LISTA DE NÚMEROS NATURALES
- 1.3 CUESTIONES
- 1.3.1 Dos comerciales en el restaurante
- 1.3.2 Invertir una lista
- 1.3.3 El palíndromo
- 1.4 RESPUESTAS A LAS CUESTIONES PLANTEADAS
- 1.4.1 Dos comerciales en el restaurante
- 1.4.2 Invertir una lista
- 1.4.3 El palíndromo

CAPÍTULO 2. COMPLEJIDAD ALGORÍTMICA

- 2.1 ESTIMACIÓN DE LOS TIEMPOS DE EJECUCIÓN DEL ALGORITMO
- 2.2 ALGORITMOS ITERATIVOS
- 2.3 ESTRUCTURAS DE DATOS

- 2.4 NOTACIÓN ASINTÓTICA
- 2.4.1 Recurrencias homogéneas
- 2.4.2 Recurrencias no homogéneas
- 2.4.3 Ejemplo de aplicación
- 2.4.4 Divide y Vencerás
- 2.5 CUESTIONES.
- 2.6 RESPUESTAS A LAS CUESTIONES PLANTEADAS CAPÍTULO 3. DIVIDE Y VENCERÁS
- 3.1 CONSIDERACIONES GENERALES
- 3.2 ROTAR UNA IMAGEN
- 3.3 BÚSQUEDA BINARIA
- 3.4 EL PROBLEMA DE LA ORDENACIÓN
- 3.4.1 La ordenación por mezcla
- 3.4.2 Ordenación rápida
- 3.4.2.1 LA BANDERA HOLANDESA
- 3.5 CUESTIONES
- 3.5.1 K-ésimo menor elemento de un vector
- 3.5.2 Pseudomediana
- 3.5.3 Rotación de una imagen
- 3.5.4 Rotación de un vector
- 3.5.5 Intercalación de dos vectores ordenados
- 3.5.6 Multiplicación de naturales grandes
- 3.5.7 Suma de la subsecuencia máxima de un vector de naturales
- 3.5.8 Moda de un vector
- 3.5.9 Búsqueda de un elemento en un vector ordenado
- 3.5.10 Elemento mayoritario de un vector
- 3.5.11 Algoritmo Shannon-Fano
- 3.5.12 Mínimo y máximo de un vector no ordenado

3.6 RESPUESTAS A LAS CUESTIONES PLANTEADAS

- 3.6.1 K-ésimo menor elemento de un vector
- 3.6.2 Pseudomediana
- 3.6.3 Rotación de una imagen
- 3.6.4 Rotación de un vector
- 3.6.5 Intercalación de dos vectores ordenados
- 3.6.6 Multiplicación de naturales grandes
- 3.6.7 Suma de la subsecuencia máxima de un vector de naturales
- 3.6.8 Moda de un vector.
- 3.6.9 Búsqueda de un elemento en un vector ordenado
- 3.6.10 Elemento mayoritario de un vector
- 3.6.11 Algoritmo de Shannon-Fano
- 3.6.12 Mínimo y máximo de un vector no ordenado
- CAPÍTULO 4. PROGRAMACIÓN DINÁMICA

4.1 PLANTEAMIENTO DEL PROBLEMA DE CONTROL ÓPTIMO

- 4.2 PROPIEDAD DE CAUSALIDAD
- 4.3 CONTROL ÓPTIMO CON PROGRAMACIÓN DINÁMICA
- 4.3.1 El problema de la producción con beneficio máximo
- 4.3.2 Solucionando problemas con Programación Dinámica
- 4.3.3 El cambio de monedas
- 4.4 CUESTIONES
- 4.4.1 El problema del cambio de monedas
- 4.4.2 Los litros de leche
- 4.4.3 Juego con fichas
- 4.4.4 Inversión en negocios
- 4.4.5 División de una cantidad
- 4.4.6 Carga de un avión
- 4.4.7 Fabricación de automóviles

- 4.4.8 Fabricación de papel
- 4.4.9 El problema de la subsecuencia común más larga
- 4.4.10 Conversión a palíndromo
- 4.4.11 Caminos mínimos en grafos dirigidos y ponderados
- 4.4.12 El problema de la multiplicación de n matrices
- 4.4.13 Multiplicar de forma óptima n matrices
- 4.4.14 Operaciones con cadenas de caracteres
- 4.4.15 Número de árboles binarios
- 4.4.16 El problema de los dos puntos más cercanos
- 4.4.17 Número de caminos distintos
- 4.4.18 La función de Fibonacci
- 4.4.19 Número de ordenaciones con los operadores = y <
- 4.5 SOLUCIONES
- 4.5.1 El problema del cambio de monedas
- 4.5.2 Los litros de leche
- 4.5.3 Juego con fichas
- 4.5.4 Inversión en negocios
- 4.5.5 División de una cantidad
- 4.5.6 Carga de un avión
- 4.5.7 Fabricación de automóviles
- 4.5.8 Fabricación de papel
- 4.5.9 El problema de la subsecuencia común más larga
- 4.5.10 Conversión a palíndromo
- 4.5.11 Caminos mínimos en grafos dirigidos y ponderados.
- 4.5.12 El problema de la multiplicación de n matrices.
- 4.5.13 Multiplicar de forma óptima n matrices
- 4.5.14 Operaciones con cadenas de caracteres
- 4.5.15 Número de árboles binarios
- 4.5.16 El problema de los dos puntos más cercanos
- 4.5.17 Número de caminos distintos

- 4.5.18 La función de Fibonacci
- 4.5.19 Número de ordenaciones con los opera

CAPÍTULO 5. TÉCNICAS DE BACKTRACKING Y VORAZ

- 5.1 CONTROL ÓPTIMO CON BACKTRACKING
- 5.2 ALGORITMOS DE BACKTRACKING
- 5.3 DISEÑO VORAZ
- 5.4 OPTIMIZAR UNA DIETA
- 5.5 CUESTIONES
- 5.5.1 El laberinto
- 5.5.2 Todas las sumas de n cifras
- 5.5.3 El cruce de calles con semáforos
- 5.5.4 El cruce de calles con semáforos (Voraz)
- 5.5.5 Oficinas de correos
- 5.5.6 El problema del cambio de monedas (Voraz)
- 5.5.7 Árbol de recubrimiento mínimo. Algoritmo de Kruskal
- 5.5.8 Expresión lógica inconsistente
- 5.5.9 Asignación de n tareas a n agentes
- 5.5.10 El problema del viajante con prisa
- 5.5.11 Horario de clase
- 5.5.12 Paradas de autobús
- 5.5.13 Paradas de autobús Voraz
- 5.5.14 Sudoku
- 5.6 SOLUCIONES A LAS CUESTIONES PLANTEADAS
- 5.6.1 El laberinto
- 5.6.2 Todas las sumas de n cifras
- 5.6.3 El cruce de calles con semáforos
- 5.6.4 El cruce de calles con semáforos (Voraz)
- 5.6.5 Oficinas de correos

- 5.6.6 El problema del cambio de monedas (Voraz)
- 5.6.7 Árbol de recubrimiento mínimo. Algoritmo de Kruskal
- 5.6.8 Expresión lógica inconsistente
- 5.6.9 Asignación de n tareas a n agentes
- 5.6.10 Subredes interesantes
- 5.6.11 El problema del viajante con prisa
- 5.6.12 Horario de clases
- 5.6.13 Paradas de autobús Backtracking
- 5.6.14 Paradas de autobús Voraz
- 5.6.15 Sudoku
- CAPÍTULO 6. RAMIFICACIÓN Y ACOTACIÓN
- 6.1 EL PROBLEMA DE LA ASIGNACIÓN
- 6.2 OTRAS MEJORAS
- 6.3 CUESTIONES
- 6.3.1 La asignación de n tareas a n agentes
- 6.3.2 Murallas
- 6.4 SOLUCIONES
- 6.4.1 La asignación de n tareas a n agentes
- 6.4.2 Murallas
- CAPÍTULO 7. ALGORITMOS PROBABILISTAS
- 7.1 CLASIFICACIÓN DE LOS ALGORITMOS PROBABILISTAS
- 7.2 FUNCIONES PREVIAS
- 7.3 LISTAS CON SALTOS ORDENADA
- 7.4 ALGORITMOS DE MONTE CARLO
- 7.5 COMPROBACIÓN DE LA PRIMALIDAD
- 7.6 ALGORITMOS DE LAS VEGAS
- 7.7 CUESTIONES

- 7.7.1 Verificación de un producto matricial
- 7.7.2 El problema de las 8 reinas
- 7.7.3 Seleccionar el k-ésimo elemento menor de un vector
- 7.7.4 Factorización de un número natural n BIBLIOGRAFÍA BÁSICA

ÍNDICE ALFABÉTICO

SINOPSIS

Tal vez éste debería ser el primer libro de programación que todo el mundo debería leer. Aquí se trata la problemática fundamental de la algoritmia: resolver un problema encontrando la solución (o soluciones) que mejor aprovechan los recursos del computador (el tiempo de CPU y el espacio en memoria).

Muchos creen que saber programar es conocer el último lenguaje de programación o los entresijos de los más potentes (C, Java, C++, entre otros). Este libro persigue romper con esa idea errónea. Saber programar es (además de buscar la implementación más eficaz) conocer las técnicas de diseño más usuales. En este libro se tratan: la recursiva, divide y vencerás, programación dinámica, backtracking, voraz, ramificación y acotación, y los algoritmos probabilistas.

Este libro le enseñará, con múltiples ejemplos desarrollados y una detallada descripción de los razonamientos seguidos en cada paso, a mejorar su capacidad para resolver y enfrentarse a un problema algorítmico.